Optimization of the track fit for the upgraded trigger

A. Oblakowska-Mucha¹, T. Szumlak¹.

¹AGH-University of Science and Technology, Krakow, Poland

Abstract

This note presents time profiling studies of the Kalman filter based track fitting software, used by the LHCb Collaboration to reconstruct charged particle trajectories, against the number of iterations of the fitter. Also, a detailed track quality analysis quantifying the impact of the number of fit iterations on reconstructed track parameters and selected physics quantities is shown. This work is a part of a large research project, still underway, that focuses on defining and implementing the next generation trigger system for the upgraded LHCb experiment that is foreseen to collect collision data beyond year 2020 (Run III). It should also be stressed that some of the ideas, regarding the upgraded trigger, have already been implemented and successfully employed during 2015 data taking (Run II).
1 Introduction

During Run I (years 2010 - 2012) the LHCb trigger system, consisting of the hardware and software parts called L0 and high level trigger (HLT) respectively, proved to be reliable and robust. With a number of improvements along the way, its performance became much better than the initial design predicted and its output rate reached approximately 4.5 kHz of events stored on tape in 2012. For Run II, which started in 2015 and will be continued till the end of the year 2018, the overall architecture of the LHCb trigger system (i.e., the two tiers structure) is the same as for Run I.

The upgraded trigger (foreseen to be operational for Run III beyond year 2020) will undergo complete modernisation. The hardware part, which is considered the bottleneck of the current one, will be removed and the capability to select interesting physics events will rely only on the high level software component. Such a flexible fully software-based system can help not only in increasing the yields but is also critical in keeping under control biases of quantities essential for selecting events that are the part of the core physics program (for instance in case of heavy flavor physics especially vital are geometrical impact parameter and particle lifetime).

The LHCb trigger group, that is working on the design of the modernised trigger for Run III, is pushing the envelope by introducing a completely new approach which is meant to tear the boundaries between the online and offline data processing. The most innovative features of the upgraded LHCb trigger are: offline quality tracking performed in the real time, run-by-run quasi online calibration and alignment as well as the trigger level physics selections. There are very interesting implications of having such a trigger system on the physics performance of the experiment. Firstly, any subsequent offline data analysis would be using the very same tracks and vertices that were used to select the events in the first place. This should assert the consistency between online and offline selections and improve time stability of the physics results in the course of data taking. Secondly, with the calibration runs performed in quasi online mode there will be no need for the frequent re-processings, which makes it possible to discard the redundant raw data completely and store only high level objects like tracks or vertices.

Although, these ideas were initially meant for the upgraded trigger system (Run III) many of them, to large extent, have already been implemented and exploited during the first year of Run II data taking. The two parts of the HLT trigger have been completely split and are working asynchronously. The L0 candidates are processed by the Hlt1 part which attempts the partial event reconstruction and selection of displaced tracks and vertices (also preliminary di-muon candidates are selected). The Hlt1 output is then stored on disks and full calibration and alignment of the LHCb detector is performed. Subsequently, the data cached in the disk buffer are processed by the Hlt2 using the updated calibration and alignment constants that are determined on run-by-run basis. The Hlt2 trigger runs the full event reconstruction, using the Kalman filter algorithm to fit particle trajectories, starting from the Hlt1 tracks and vertices and on top of this performs the particle identification. The results obtained by studying the performance of

\[\text{The HLT is, in turn, divided into two parts called Hlt1 and Hlt2}\]
the online executed tracking (no offline reprocessing) showed that this approach is viable and was already accommodated for the current Run II. Moreover, the quality of tracks reconstructed at trigger level allowed to run successfully the offline like event selections which resulted in data samples of high purity. For instance, the real time physics selections have been performed for various charm decay channels, which allowed a rapid cross-section measurements to be performed and published as the first results with 13 TeV collision data [3].

The first topic of our project was to measure the time profile of the overall HLT tracking sequence as a function of the number of iterations that is used to estimate the final trajectory of a reconstructed particle. The main question being whether it is possible to "squeeze" the more complicated (so to say, more offline like) fitter into the severely limited time budget of the online system. The second part of this project was dedicated to detailed studies of the track quality obtained with this simplified fitting scheme. This has allowed us to understand and measure directly its impact on the reconstructed track parameters and eventually on the physics performance of the LHCb detector. Although the idea of running the online tracking with the simplified detector model was initially meant for the upgraded trigger we showed that it is already possible to use the same principles for the current system. All the results presented in this note are produced using the software platform of the present LHCb trigger and dedicated Monte Carlo signal samples containing $B^0 \rightarrow K^* \mu\mu$ decays.

The remainder of this note is organised in the following way: in Section 2 a description of relevant Kalman filter features is presented, Section 3 describes time profiling of the Kalman filter algorithm used to fit tracks, a detailed studies of the track quality for different number of the Kalman algorithm iterations are presented in Section 4 which is followed by a short Summary. Throughout this document, natural units are used.

2 Kalman Fitter

In this section we provide an abridged description of the track fitting procedure used by the LHCb reconstruction software. We are not going to delve into technical aspects of the algorithm implementation but rather give an overview of the fitting process as a series of consecutive steps that are necessary to find a particle trajectory and estimate its physical properties such as the momentum.

The Kalman based track fitting algorithm is run in the software trigger (HLT). The Kalman filter method consists of multiple repeated steps that enable the evaluation of track parameters. In order to improve its quality it is possible to identify and remove outlier hits (i.e., the ones with the highest contribution to the track’s $\chi^2$). Subsequently, the track is fitted without them. During the entire track parameters estimation procedure up to two outlier hits may be removed. The fitting procedure was the most time consuming part of the track reconstruction sequence during Run I. Therefore, for the online reconstruction, the fitting algorithm is using a series of approximations such as the simplified detector

\textsuperscript{2}See Section 2 for details.
Geometry [6], lite clusters [7], and fewer iterations - actually the initial configuration allowed only one bi-directional execution of the Kalman filter. In particular, the simplified detector model was essential in enabling the track reconstruction to fit into the limited time budget of the LHCb software trigger. The basic idea behind the simplified geometry is to provide the detector model using a small number of simple geometrical volumes (e.g., cylinders or cuboids) representing the respective sub-detectors, in contrast to the full geometry, that describes the spectrometer in great detail [8]. In addition, since we describe entire complex sub-detectors with single volumes, they also need to be filled with specially composed imaginary "effective" materials (see also Section 3.1). The optimisation process that allowed to obtain the material content of the simplified volumes was critical from the point of view of the physics performance. Appropriate material description is necessary for the correct estimation of the multiple scattering and energy loss effects, that in turn, are of the paramount importance for the tracking reconstruction procedure [9].

The Kalman filter used by LHCb tracking software takes as the input a collection of hits reconstructed by tracking detectors and fits them to a model based on the equations of motion in a complicated inhomogeneous LHCb magnetic field (it can be done using numerical methods and the full field map) and taking into account particle multiple scattering and energy loss in detector material. The implementation of the Kalman filter used by the LHCb consists of two stages: bi-directional filtering and smoothing. The first stage runs an algorithm that iterates over the input detector measurements (nodes) performing prediction and projection steps in order to determine the best track parameters estimate at a current node. Let’s assume that we have \( k \) nodes in the input container and wish to proceed from node \((j-1)\)-th to \( j \)-th. First, the prediction is made by propagating the track segment (determined using the previous \((j-1)\) nodes) through the detector taking into account particle interaction with the material. Next, the projection is made by determining the residual value and its uncertainty between the predicted track state and the \( j \)-th measurement. This is then used to update the predicted track state and perform the next prediction step to the \((j+1)\)-th node. When all track nodes are used, the second filtering is performed in the opposite direction. Finally, the filtered states at each node are smoothed by averaging the results. Both filtering steps and the smoothing form a track fit iteration.

Track fit iterations are repeated as many times as necessary (the maximum number of iterations for the LHCb tracking software is set to 10), in order to make it converge which should result in the optimal track state parameters. As a measure of compatibility between the hits and the track model the \( \chi^2/n_{\text{dof}} \) is calculated and if it changes less than 0.1 in the next iteration, the track is considered as converged in the previous one. After smoothing, the algorithm can reject measurements that were incorrectly added to this

---

3VELO clusters represent charged particle hits in the software. They contain information regarding the local position of a hit (expressed as a respective read-out channel number), the number of contributing read-out channels and the signal amplitudes registered on them. In tracking software we use two different types of VELo clusters: lite and full.

4The simplified geometry contains approximately 30 volumes, whilst the full one around one million.

5This is obviously true regardless of the type of geometry that is used in the track reconstruction.
track. Such an outlier is removed from the fit if its $\chi^2$ contribution to the fit is too large. Once this happens the fit must be done once more to provide accurate track parameters without this measurement. From the above remarks we can conclude that the total time of Kalman filter execution depends not only on the type of detector material description but also on the criteria used for fit convergence and especially on the maximum number of iterations which can be performed to obtain optimal track parameters.

3 Timing Studies of the Kalman Filter

In this part of the note a detailed description of the time profiling of the Hlt2 fitter is given. First, we describe the software used in this analysis and all the metrics we chose to measure the time performance of the fitter. Next, the results and their discussion is presented. Since the bulk of the analysis was done before the start of the Run II, in the following text we make a clear distinction between the online and offline tracking. Such distinction is no longer valid for the current operation (Run II) because the track reconstruction is done the same way for both online and offline applications. We decided to keep this distinction to underline the differences between configuration settings of the offline fitter used in Run I and the current one.

3.1 Description of the Measurements Setup

As already mentioned in the previous section, the HLT fitter uses a series of approximations to run faster. In particular the most important differences between the online fitter and its offline counterpart are: simplified geometry, lite clusters and reconstruction of the particle trajectories with a single Kalman filter iteration. Thus, first of all we include in our studies two scenarios where both simplified and full geometry are used for increasing number of fit iterations. For the sake of this paper the significance of using either is shortly described in the following text. The critical part of the track fitting procedure is locating intersections of particles with the material of the detector within its angular acceptance. For the offline case this is done using the detailed detector model (i.e., the full geometry) which contains exact description of all detector elements (active or dead) along with their actual material properties. Since the full geometry contains approximately $O(10^6)$ volumes in a treelike structure, the calculation of a given intersection may be slow. On the other hand the simplified geometry contains $O(10)$ elements filled with the effective material giving approximately the same material budget as the real detector. In this case the intersection determination is fast, since it does not involve time consuming browsing of the complicated hierarchy of the detector volumes.

Apart from the full geometry the offline fitter makes use of the full clusters that allows a better estimate of the particle’s position and its uncertainty. Both these quantities are given in terms of a detector channel (i.e., a local position) that can be subsequently translated into the global spatial position. Our studies showed that changing from the full to lite clusters has a small impact on the timing results and we decided not to include
them in this note. The major difference between these two types of hit classes boils down to the additional information regarding the charge deposited on the channels contributing to a given hit that is included in the full clusters. It seems that the tracking procedure does not gain much, if at all, by including a more precise estimate of the particle position using the full clusters. The information carried by the lite clusters is sufficient to get a performance that is the same as in the case where the full ones are employed (this holds also for the results presented in Section 4). Similarly, varying the maximal number of outliers removed in the fit did not change visibly our results.

3.2 Timing Metrics Definitions

In this section the results of the time profiling of the Kalman filter are presented. We decided to quantify them using the following variables: $T_{\text{Kalman}}^{(i)}$, $R_{T_{\text{Kalman}}}^{(i)}$, $T_{\text{Kalman/Hlt2}}^{(i)}$, and $T_{\text{Kalman/Tot}}^{(i)}$ (see Table 1). The first one represents the average (per event) absolute time used for track fitting with the Kalman filter in the Hlt2 trigger for a given number of iterations ($i$). Next, we decided to show the ratio of the time spend on track fitting as a function of the maximum number of iterations, to the time of a single iteration:

$$R_{T_{\text{Kalman}}}^{(i)} = \frac{T_{\text{Kalman}}^{(i)}}{T_{\text{Kalman}}^{(1)}}$$

Finally, we also use two metrics showing the relative contribution of the track fitting procedure with respect to the total time of the Hlt2 and HLT trigger respectively, again determined as a function of the maximum number of iterations. All these numbers are given for both simplified and full detector geometries. Results and their discussion are presented in the next section.

3.3 Results

First of all, using the reconstruction software available at the time of the analysis (i.e., Brunel v48r2) we noticed a substantial difference in execution time depending on the type of detector model, full versus simplified, used by the fitter (see Figure 1 and Table 2). The difference amounts to 300% and practically does not depend on the number of iterations of the Kalman fitter. This feature is very well understood: the full geometry contains approximately one million volumes, which makes the calculation of the track intersections with detector material time consuming.

During our studies we also ran the Moore application using selected HLT test nodes. We noticed quite substantial differences in the execution time between the nodes, that amounted to 80%. However, this is expected since the main difference between the used nodes is related to the CPU chips installed. This time variation has not been studied in more detail since we were interested primarily not so much in the absolute value of

---

6One should note a caveat of measuring also the overhead coming from the wrapper algorithm that actually calls the fitter.
Table 1: Definition of the variables used to quantify the time profiling procedure of the Hlt2 Kalman fitter. The maximum number of iterations is denoted by $i$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T^{(i)}_{Kalman}$</td>
<td>Absolute time of the Hlt2 Kalman fitter</td>
<td>ms/evt</td>
</tr>
<tr>
<td>$R^{(i)}_{Kalman}$</td>
<td>The ratio of the $i$ iterations execution time to a single iteration time</td>
<td>-</td>
</tr>
<tr>
<td>$T^{(i)}_{Kalman/Hlt2}$</td>
<td>The ratio of the execution time of the $i$ iterations to the execution time of the Hlt2 trigger</td>
<td>%</td>
</tr>
<tr>
<td>$T^{(i)}_{Kalman/Tot}$</td>
<td>The ratio of the execution time of the $i$ iterations to the total execution time of the HLT trigger</td>
<td>%</td>
</tr>
</tbody>
</table>

Figure 1: Absolute time per event used by the Hlt2 Kalman filter algorithm plotted as a function of the maximum number of iterations. Full circles and triangles correspond to the simplified and full detector geometry descriptions respectively.

Table 2: The change in absolute time of execution of the Kalman filter, $T^{(i)}_{Kalman}$, with the increasing number of iterations for full and simplified geometry.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>full geometry</th>
<th>simplified geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2</td>
<td>16%</td>
<td>25%</td>
</tr>
<tr>
<td>2 → 3</td>
<td>1.7%</td>
<td>14.8%</td>
</tr>
<tr>
<td>3 → 4</td>
<td>4.5%</td>
<td>9%</td>
</tr>
<tr>
<td>4 → 5</td>
<td>2%</td>
<td>1.5%</td>
</tr>
</tbody>
</table>
the execution time as in its relative increase as a function of the maximum number of fit iterations. In addition, almost all working nodes use the faster chips anyway.

In order to quantify in a convenient way the time consumption increase as a function of the maximum number of Kalman iterations we introduced the variable $\frac{T_{i_{\text{Kalman}}}^{(i)}}{T_{i_{\text{Kalman}}}^{(1)}}$, which is calculated as the ratio of the execution time taken by the reconstruction for a given number of iterations $i$ and that consumed by a single iteration. Appropriate profiles representing the relative increase in execution time have been made for both types of detector geometry. As can be seen in Figure 2 (see also Table 3) both curves have very similar features: there is sharp jump between a single and two iterations and then the difference gradually saturates: there is a negligible variation above six iterations. We noticed one more interesting point: although the total execution time for the full geometry is much longer (in absolute units) than for the simplified one, the profile plots show that the relative time difference for the former are almost flat beyond two iterations, whilst for the latter case they show a considerable gradient up to the fifth iteration. For the simplified model two iterations take approximately 27% more time than a single one, while

Table 3: Increase of the execution time of the Kalman filter, $\frac{T_{i_{\text{Kalman}}}^{(i)}}{T_{i_{\text{Kalman}}}^{(1)}}$, with the increasing number of iterations for full and simplified geometry.

<table>
<thead>
<tr>
<th>Iterations</th>
<th>1 → 2</th>
<th>2 → 3</th>
<th>3 → 4</th>
<th>4 → 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>full geometry</td>
<td>19%</td>
<td>8%</td>
<td>4.1%</td>
<td>3.2%</td>
</tr>
<tr>
<td>simplified geometry</td>
<td>27%</td>
<td>16%</td>
<td>6.7%</td>
<td>2.5%</td>
</tr>
</tbody>
</table>
Figure 3: The ratio of the time taken by $i$ Kalman filter iterations to the time of the Hlt2, $T_{\text{Kalman}/\text{Hlt}2}^{(i)}$, (upper plot) and to the total execution time of Hlt, $T_{\text{Kalman}/\text{Tot}}^{(i)}$, (lower plot) per one event.

the same figure for the full geometry is 19% (see Table 3).

Finally, we can look at the contribution of the fitting procedure to Hlt2 and the total HLT execution time respectively, for both the simplified and full geometry, expressed as a function of the number of used iterations $i$ (see Figure 3). In case of the simplified model its relative contribution to Hlt2 and HLT execution times varies in range 12-17% and 9-13% respectively. In turn, for the full detector description we obtained 30-33% and 24-26%.
4 Track Quality Studies

This section reports on the statistical analysis of the performance of the track fitting procedure. We studied the change in track parameters using both the full and simplified detector description as a function of number of the Kalman filter iterations.

We compare selected quantities, that can be used to describe and evaluate the tracking performance: momentum, momentum resolution, slopes and their resolutions, total $\chi^2$ and reduced $\chi^2/ndof$. We decided to follow two different approaches. The first one is just a simple statistical analysis of the listed quantities using all tracks produced by the reconstruction software for a selected maximum number of fit iterations. For the second, we perform a similar analysis but we compare parameters for exactly the same tracks fitted with increasing number of iterations and with both types of detector geometry description. Only long tracks\(^7\) were used to perform these analyses.

4.1 Statistical Analysis of Selected Track Properties

The default settings for the offline reconstruction require full geometry detector description and a maximum of ten iterations of the Kalman filter. The distribution of the considered track parameters are shown in Figure 4. The momentum resolution is represented by:

$$\frac{\delta p}{p} = \frac{1}{p_{\text{rec}}} - \frac{1}{p_{\text{true}}}.$$

An average value of 0.35% is obtained which agrees with the figures cited in other analyses. The residual of the slope $t_x = \frac{dx}{dz}$, for a given track, is calculated as a difference between the slope taken at its first state and the corresponding Monte Carlo true information: $t_{x,\text{rec}} - t_{x,\text{true}}$. The slope resolution is measured to be 0.02%. The histograms were fitted with a double Gaussian model to quantify the respective resolutions and to check for possible biases. Both distributions are well centered at zero (there is a negligible offset of the order of $10^{-5}$ and $10^{-7}$ for momentum and slope resolution respectively) indicating that there is no systematic shift.

In the further study we investigated the differences in these parameters that appear after reconstructions with various settings (i.e., different detector models and fit iterations). A small discrepancy in the momentum resolution occurs for low momentum tracks, which is shown in Figure 5. The dependence of the momentum resolution and bias on the momentum are shown for a single and eight iterations respectively. In both settings a minimum in the resolution distribution is located around 3 GeV. This minimum is expected due to the multiple scattering $1/p$ dependence that plays a dominant role for low momentum tracks. For higher momentum tracks the error on position measurements (which is proportional to $p$) starts to play a dominant role and so the resolution is slightly worse.

Further analysis showed that if we change the number of the maximum of Kalman filter iterations, there is a negligible difference in the momentum distribution between any configuration applied. The momentum resolution distributions are slightly different between

\(^7\)A long track can be assigned only to a particle that is registered by all tracking sub-detectors.
Figure 4: The track parameters obtained with the default offline settings: momentum (top left-hand side), momentum resolution (top right-hand side), slope $dx/dz$ (middle left-hand side), residual distribution of $dx/dz$ (middle right-hand side), $\chi^2$ value for all tracks (bottom left-hand side) and $\chi^2/ndof$ for all tracks (bottom right-hand side).

one and two iterations. The largest discrepancy is observed for the low momentum tracks (below approximately 4 GeV). Any differences practically disappear after three or more iterations, see Figure 6. The same behavior is observed in both full and simplified geometry description. Also, we noticed that there is hardly any difference in the estimated track parameters obtained using the full or lite clusters. This result is somewhat counterintuitive, since one would expect that having a much more precise estimation of the cluster position uncertainty ought to result in better track parameters performance. Although interesting, this problem is out of the scope of this project and should be studied in more details in a

---

8See the description of the full and lite clusters given in Section 2.
Figure 5: The bias in the momentum resolution as a function of the momentum (top left-hand side), with a detailed look at low momentum region (top right-hand side). Analogous distributions for the momentum resolution $\frac{\delta p}{p}$ (bottom left- and right- hand side). Tracks are split with respect to their charge and a single iteration is compared with the eight iterations case.

dedicated analysis.

Figure 6: Comparison of the momentum resolution of all tracks reconstructed with one and two Kalman filter iterations (left-hand side) and two and four iterations (right-hand side). In both cases the full geometry description of the detector was used.

Comparison of the $\chi^2$ and $\chi^2/ndof$ leads to a similar conclusion – a significant difference is noticeable only between the reconstruction performed with one and two iterations (see Figure 7), and increasing the number of iterations does not change these parameters by
much.

The number of tracks that we gain by running more iterations amounts to approximately 2.5%, which is the number of additional tracks we see after eight iterations with respect to a single one. Showed results on momentum resolution for full and simplified geometry are summarised in Table 4. Regardless of the number of fit iterations there are only negligible differences between track parameters reconstructed with full or simplified geometry (Figure 8).

![Graphs and charts showing chi-squared distributions for different iterations and geometries.](image)

Figure 7: The $\chi^2$ (top plots) and $\chi^2/\text{ndof}$ (bottom plots) distributions obtained for one and two (lef-hand side plots) and two and four (right-hand side plots) Kalman filter iterations using the simplified detector geometry description.

<table>
<thead>
<tr>
<th>Number of iterations</th>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>full geometry</td>
<td>$\sigma_p$ $[10^{-3}]$</td>
<td>3.69</td>
<td>3.52</td>
<td>3.52</td>
<td>3.51</td>
</tr>
<tr>
<td>simplified geometry</td>
<td>$\sigma_p$ $[10^{-3}]$</td>
<td>3.77</td>
<td>3.61</td>
<td>3.61</td>
<td>3.60</td>
</tr>
</tbody>
</table>
4.2 Track by Track Comparison

A much more interesting analysis deals with the comparison of the parameters of the same tracks but reconstructed with different settings. Again, we compare the full and simplified detector geometry description and change the number of Kalman filter iterations. The procedure goes as follows. First, a reconstruction is done on the same set of tracks, but with the two different settings. Then, we look for the same tracks in both samples and compare their reconstructed parameters. In such a way we can compare full and simplified detector description or investigate how the next Kalman filter iteration improve (or influence at all) the track parameters. The results are presented in Figures 9 - 13.

There is a small improvement in the momentum resolution when the second filter iteration is performed, but further ones do not change its value (see Figure 9). The slight difference between the full and simplified geometry description is observed for any number of iterations, but the observed variation is almost constant and negligible in value (see Figure 10). The comparison of the momentum resolution for different settings is given in Table 5.

Table 5: The momentum resolution obtained in the track by track analysis for full and simplified geometry and selected number of iterations $i$.

<table>
<thead>
<tr>
<th>Number of iterations</th>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>full geometry</td>
<td>$\sigma_p$ $[10^{-3}]$</td>
<td>3.55</td>
<td>3.39</td>
<td>3.38</td>
<td>3.38</td>
</tr>
<tr>
<td>simplified geometry</td>
<td>$\sigma_p$ $[10^{-3}]$</td>
<td>3.64</td>
<td>3.49</td>
<td>3.49</td>
<td>3.49</td>
</tr>
</tbody>
</table>

We can also check the impact of a previous iteration on the subsequent ones. To do so, for each track we take its momentum obtained for a given reconstruction settings (for instance simplified geometry with one iteration) and subtract it from the momentum estimated for this track with the same settings but with a different number of iterations.
Figure 9: Comparison of the momentum resolution of the same tracks reconstructed with one and two (left-hand side) and two and four (right-hand side) Kalman filter iterations. Top and bottom plots correspond to simplified and full geometry description respectively.

(e.g., simplified, two iterations). In this way we can determine what is the fraction of tracks that acquire their final momentum estimate after a given number of iterations as well as how much the momentum value changes for tracks that are re-fitted in next iterations. The results are presented in Figure 11. One can see that after two iterations there are approximately 1.2% of tracks for which the momentum estimate differs by more than 500 MeV. The detailed study showed that such a large change appears for higher momentum tracks and when we investigated the relative change of momentum between iterations, the results (presented in Figure 11) show that lower momentum tracks need more iteration to obtain their final momentum value. For all analysed tracks after four iterations only negligible changes in the momentum are observed. The track $\chi^2$ and $\chi^2/ndof$ obtained with different setting are shown in Figure 12. It is visible that the second iteration reduces $\chi^2$ of the fit, but the next one performed has no significant influence.

An interesting effect can be seen when we check how the next fit iteration change the $\chi^2$ of the tracks fit. In the second iteration more tracks have lower $\chi^2$ value (see Figure 13) but after four iterations this distribution is symmetric, indicating that on average, the same (and small) number of tracks are reconstructed with both somewhat larger and somewhat smaller $\chi^2$.

By adjusting appropriately the software configuration we were able to run the track fitting without outlier removal (by default up to two outliers can be removed using a $\chi^2$
Figure 10: Comparison of the momentum resolution of the same tracks reconstructed with different geometry description. Plots correspond to one (top left-hand side), two (top right-hand side), four (bottom left-hand side) and eight (bottom right-hand side) Kalman filter iterations.

Figure 11: The change in momentum measured for tracks reconstructed with an increasing number of Kalman filter iterations (left-hand side.) The relative $\frac{|\Delta p|}{p}$ change in momentum between different settings as a function of momentum (right-hand side).

cut). The results are presented in Figure[14]. In the case where no outlier removal is allowed, the $\chi^2$ values tends to have noticable higher values. In addition, we showed on the right-hand side plot in Figure[14] the difference between the $\chi^2/ndof$ evaluated for the same tracks after eight iterations. The resulting distribution is strongly biased towards negative values, which reflects quantitatively the impact of the outliers removal on the track’s $\chi^2$. 
Figure 12: Comparison of the $\chi^2$ and $\chi^2/ndof$ of the same track reconstructed with one, two and four Kalman filter iterations.

Figure 13: The difference between the $\chi^2$ (left-hand side plot) and the $\chi^2/ndof$ (right-hand side plot) values of the same track but reconstructed with an increasing number of Kalman filter iterations. The distribution is obtained by subtracting the values of different settings: for example $\chi^2(1 \text{ iteration}) - \chi^2(2 \text{ iterations})$ when we compare one, two, four and eight iterations.

5 Summary

In this note we present studies regarding the Kalman filter algorithm used to fit track trajectories reconstructed by the LHCb detector. The main topic of this work is related to the differences between the so called online and offline modes, that mainly boil down to a smaller number of fit iterations and simplified detector model. The motivation behind this
analysis is related to a novel idea of performing the final tracking in the online system using the real-time calibration information. In this way the Collaboration uses the same tracking software throughout the entire data taking period and there is no longer a need for time consuming track reprocessing campaigns. Since the online systems perform their job in real time, bringing the HLT tracking performance closer to the offline one must be based on some compromises, i.e., use less iterations and use a much simpler geometry and material description. Our analysis was dedicated to study the timing of the Kalman filter as a function of the maximum number of iterations and to check the overall performance of the tracking obtained for different number of iterations. The results of this study showed that using the simplified geometry and fewer fit iterations do not compromises the track quality, we see just tiny differences in case where four or more iterations are applied. For Run II data taking (started in 2015) it was decided to use the simplified geometry and a maximum of ten Kalman fit iterations for track reconstruction purposes.

The timing studies, performed with selected HLT working nodes, showed a specific time consumption profile (see Figure 1) that depends primarily on the type of geometry model and increases as a function of the number of fit iterations. In case of the full description one can notice that the time used by the fitter saturates almost immediately, i.e., above three iterations. For the simplified geometry, the saturation plateau is reached after the fifth iteration; however, differences in the execution time are still present for the consecutive iterations. The timing analysis showed that the time consumption for increasing number of Kalman iterations is under control and does not blow up when we change from one to ten iterations. The relative rise in the execution time between one and ten iterations amounts to about 30% and 70% for the full and simplified geometry respectively. It should be borne in mind that the absolute time of execution is around three times longer when the full detector model is used, so for each number of iterations the execution time of the tracking procedure using simplified description is always a small fraction of that consumed by the same code using full description.

The statistical analysis of the selected quantities used for measuring the quality of
reconstructed tracks yields a very interesting results. Apparently the momentum and track slopes and their respective resolutions do not change much from iteration to iteration. Also, the difference in the performance between the full and simplified detector geometry and material models are small and practically disappear after the second iteration. The $\chi^2$ distributions plotted for the increasing number of iterations show that fitted trajectories converge quickly for a large fraction of tracks, and for the rest there is a small change: some of the tracks feature smaller, and some of them larger, values of the $\chi^2$. The plots that present the difference in the $\chi^2$ value calculated for tracks after each iteration are symmetric, suggesting a small random migration (see Figure 13). Also, there seems to be no appreciable impact on the tracking performance depending on the type of clusters used: there is no degradation in the momentum or slope resolution with lite clusters.

References

[6] M. Gersabeck, E. Rodrigues, Tracking and physics validation studies of the simplified geometry description with $B^0_{(s)} \to h^+h^-$ decays, LHCb-2008-030